

TRACKIT APPLICATION

Project report submitted in partial fulfilment of the requirement for the degree of
Bachelor of Technology

In

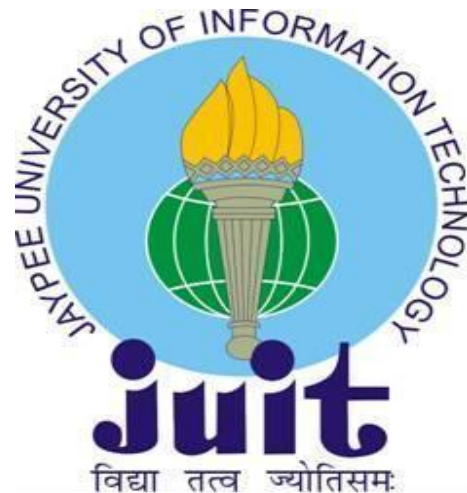
Computer Science and Engineering/Information Technology

By

Aman Prajapati (171461)

Under the supervision of

Mr. Rizwan Ur Rehman



Department of Computer Science & Engineering and Information Technology
**Jaypee University of Information Technology Wagnaghat, Solan-173234,
Himachal Pradesh**

Certificate

Candidate's Declaration

We hereby declare that the work presented in this report entitled “**TRACKIT APPLICATION**” in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering/Information Technology** submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology Waknaghat is an authentic record of our own work carried out over a period from January 2021 to May 2021 under the supervision of **Mr. Rizwan Ur Rehman** (Assistant Professor(Grade II), Computer Science and Engineering And Information Technology)

The matter embodied in the report has not been submitted for the award of any other degree or diploma.



Aman Prajapati, 171461

Certificate

Certificate by Supervisor

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

A handwritten signature in black ink on a white background with horizontal lines. The signature is stylized and appears to be 'Rizwan Ur Rehman'.

Mr. Rizwan Ur Rehman

Assistant Professor (Grade II)

Computer Science and Engineering and Information
Technology

Dated:

Project Report Undertaking

I Mr. /Ms. Aman Prajapati

Roll No. 171461

Branch Computer Science and Engineering is doing my internship with Infosys Limited from 1st February 2021 to 13th May 2021

As per procedure I have to submit my project report to the university related to my work that I have done during this internship.

I have compiled my project report. But due to COVID-19 situation my project mentor in the company is not able to sign my project report.

So I hereby declare that the project report is fully designed/developed by me and no part of the work is borrowed or purchased from any agency. And I'll produce a certificate/document of my internship completion with the company to TnP Cell whenever COVID-19 situation gets normal.

Signature:



Name: Aman Prajapati

Roll No.: 171461

Date: 20th May 2021

ACKNOWLEDGEMENT

In the present world of ever evolving technology and cut throat competition in every field, there is a race of existence in those having the will to come forward and succeed. Most importantly, we would like to express our sincere gratitude to our advisor Mr. Rizwan Ur Rehman for the continuous support in the study and research. It was under his guidance and effort that we were able to implement this project successfully.

Beside the hard work put by the team and the mentor, we would also like to express our gratitude to the panel for bestowing us with an opportunity to present this project and providing us with feedback which will assist us in future.

We would also like to extend our acknowledgement to our esteemed institute Jaypee University of Information Technology for exposing and motivating us to work in various fields.

TABLE OF CONTENTS

Chapter	Page
1. Introduction.....	8-11
1.1Introduction.....	8
1.2Objectives.....	9
1.3Problem Statement.....	9
1.4Technology Used.....	10
1.5Languages Used.....	11
2. Literature Survey.....	12-40
2.1Full Stack Web Development.....	12
2.2Front End Framework.....	20
2.3Back End Framework.....	30
3. Implementation.....	41-47
4. Result.....	48
5. References.....	49

List Of Figures

Figures	Pages
Figure 1	7
Figure 2	8
Figure 3	10
Figure 4	12
Figure 5	17
Figure 6	18
Figure 7	20
Figure 8	28

ABSTRACT

In this venture, we got an outline of what a full stack engineer is and what full stack web development involves. We likewise found out about the design of current web applications. We got an outline of AngularJS as a front-end structure and Spring Boot as a backend system. We fostered our first Spring Boot application. We took a gander at the idea of single-page applications. We carried on with the existence pattern of AngularJS and saw various stages, for example, the bootstrap stage, the aggregation stage, and the runtime information restricting stage. At that point we find out about AngularJS engineering ideas and the MVC design. We set up the AngularJS improvement climate by adding AngularJS to the Spring Boot application we created in. We utilized Bootstrap for CSS planning.

We saw various segments of AngularJS and executed the model, view, and regulator in our application. At that point we made distinctive view pages to burn-through your REST API. At long last, we have figured out how to screen our Spring Boot application utilizing an actuator.

CHAPTER 1

(INTRODUCTION)

1.1 INTRODUCTION

With the fast development in computer vision area, more and more research works and industry applications are focused on different applications. Providing different features and easy access to users would act as a fundamental part of many applications, including TrackIt application. TrackIt Application is a company application which will be used by the company employees for their regular robust functioning and dealing with daily assigned deeds. Thus, the major objective of TrackIt Application is to serve employees in a fundamentally objective and appropriate manner.

There are two main challenges for TrackIt Application, one is that it should be easily accessible to all the employees irrespective of the posts they hold and responsibilities they carry for serving better and optimistic job idealistic goals for the company the other is that we have to work on new and challenging technologies with keeping a stand for further improvements and changes.

Our primary motivation for this project was the responsibilities assigned by our project mentor and supervisor and batch owner of the company in which we were offered training for a three months period. The applications of this research are numerous and significant. The idea we are working towards solving an open and important problem with all these possible applications greatly inspired us. The competition element also allowed us to benchmark our results against the greater community, and compare the effectiveness of our methods against alternatives. Lastly, we were motivated by

the inherent challenges associated with the problem. Our motivation lead us too believe it would be a deep and interesting topic.

1.2 OBJECTIVES

In recent Covid times, it is seen the task efficiency is at challenge for all the employees out there. We intend to apply the courses, we have learnt in the training phase of combined three months to practical application. We intend to help serve a completely online based educational system which not only helps the educators and team of the task handlers but also the trainees who completely rely on the online learning system for this covid period.

In this project, we help to provide an application that serves as a assistant in managing and scheduling different sessions at appropriate timings to different educators and co-educators out there.

1.3 PROBLEM STATEMENT

They gave us three problem statements to implement for the project as it was a working project used by the students and educator for which we have to develop 3 different problem statements which will be added to the actual working project, the problem statements were-

1. As a trainee/batchowner one should be able to login to our respective pages automatically just by entering the user id.
2. As a trainee one should be able to view our personal details along with our academic performance.
3. As a trainee one should be always connected to our batchowner directly.
4. As a trainee one should be able to put queries to the batchowner and also view the past queries put by me.
5. As a Batchowner one should be able to view my profile.
6. As a Batchowner one should be able to view the details of trainees under me.
7. As a Batchowner I should be able to see the queries put to me by my trainees and reply to them.

We have to implement all the problem statements in two sprints because we were using the agile model for implementing the project so we divided the project into sprints so we get two sprints of 14 days and 10 days. We have to use the technologies we studied in the training like (Java, Spring Boot, MongoDB, Angular, and Bootstrap) and implement the project in group using github for the project so that everyone can contribute to the project easily and progress of the work be done.

1.4 TECHNOLOGY USED

In making a TrackIt Application, we need a database, a frontend that will interact with the user and an API that will communicate and transfer information between the frontend and the database.

- For developing the frontend, using Angular.
The front of the application is made angular as it offers a variety of templates and layouts to make the app more attractive.
- For API implementation, Boot Spring is used.
After the application it was designed using the Spring Framework which is widely used in CRUD operations to develop a business application.
- MongoDB database is used to store data.
- The tools used to improve the TrackIt Application query are:
 1. Angular VS code.
 2. Spring Boot Eclipse.

1.5 LANGUAGES

We have used Java including its libraries and we have also used the spring framework that Java supports in building the API. We used angular to build the front of the app.

FRONTEND

1. CSS
2. HTML
3. TypeScript
4. Bootstrap

BACKEND

1. Java

CHAPTER 2

LITERATURE SURVEY

2.1 Full Image of Full Stack Web Development

Full web improvement permits you to test the best instruments and systems while making your application vigorous and dependable in different angles like form, toughness, sturdiness and security.

In this part, I will cover the bigger image of the full improvement of the web and examine the production of present day applications to beat the legend that improving work area applications and applications is a decent way. I will discuss AngularJS as the front edge and Spring Boot as the last casing. I will likewise tell you the best way to set up Spring Boot as an advancement site. After that you will overhaul your first Spring Boot program and utilize this program. At last, you will screen your Spring Boot framework utilizing the Spring Boot actuator.

2.1.1 FULL Stack Developer

At whatever point somebody reveals to you that you are a full stack engineer, the primary inquiry you will pose is, "The thing that sort of full stack developer are you?" The thing is, in the event that you work with the Web today, you need to utilize a ton of apparatuses. The stack has expanded than it was previously. Organizations these days like to recruit engineers who are adequately adaptable to do anything as opposed to have a great deal of skill in only one innovation. A definitive objective is to have the option to code in Java, utilize Angular, compose information scripts, compose CSS, etc. Indeed, a couple of years prior, it was not difficult to say that you were a finished stack producer, yet today you should deal with Java, AngularJS, Bootstrap, CSS, information contents, and more subtleties, and you should think about content

conveyance, posting, and how to distribute to the assortment of Amazon Web Services. Obviously, it is confounded to deal with the entirety of this without a moment's delay. What's more, I've never at any point discussed zones like planning the UI and making versatile applications. In this way, there is a ton you need to know when you say you are the maker of the ideal stack. Is the stack-filled specialist the person who knows the most un-pretty much every one of these subjects or is the most flexible? Time after time, undeniable stack engineers are viewed as jacks-of-all-exchanges yet are experts of nothing. Subsequently, it is smarter to choose what sort of full-time engineers.

Being a full stack creator implies you should be a connector. Thus, regardless of whether you just have one bunch of good abilities, you can set up your work for conveyance so another person can manage their work on the opposite side and expertise to get it. This is the place where full stack engineers are generally significant. However, the odds of tracking down a decent stack engineer are very low. The thing is, new businesses consistently need a full stack manufacturer who can care for everything, and new companies are consistently anxious to recruit full-stack designs at any expense. The full stack motor designer in the business is a specialist engineer in everything from customer side composing like HTML, CSS, and AngularJS to worker side like Java and Spring Boot and has data on an assortment of subtleties, including both RDBMS and NoSQL information base projects . In this manner, an undeniable website designer ought to have a decent review insight and client experience with the worker side

A full stack designer is liable for making all pieces of the framework work appropriately. The designer should have aptitude in the accompanying:

1. Creating user connections Creating business logic and service layers on the server side.
2. Handling relationship-related and unrelated information
3. Manage API communication
4. Protect systems using authentication and systematic authorization

5. Ensuring quality assurance
6. Understanding the needs of the business and the needs of the customers.

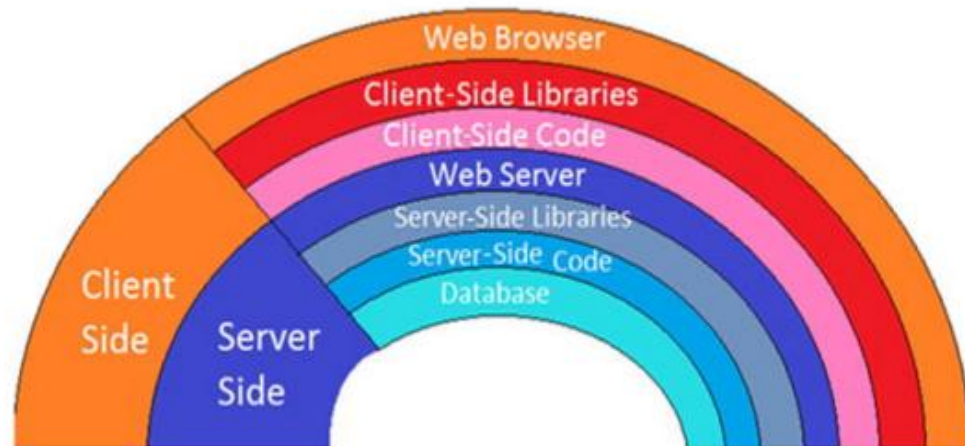


Figure 1: Alternative technology for customers and servers

2.1.2 FULL STACK DEVELOPMENT

A stack is a mix of apparatuses and components that make up an application. In a web application, the parts can be conclusive (HTML, CSS, JavaScript/jQuery) or foundation (worker execution, sensible layers, and subtleties). Note that today the information base can be ordered rather than worker side execution because of information as a help (DaaS). Full web advancement implies that you are open to working with both front and back innovation. Basically, a total stack advancement incorporates client side improvement utilizing HTML/CSS, JavaScript/jQuery, and a structure like AngularJS and Bootstrap. In worker side turn of events, it incorporates programming dialects like Java, systems like Spring Boot, information like PostgreSQL or Oracle or another data set, and everything in the middle. The association between the customer side and the worker side happens through RESTful API and HTTP.

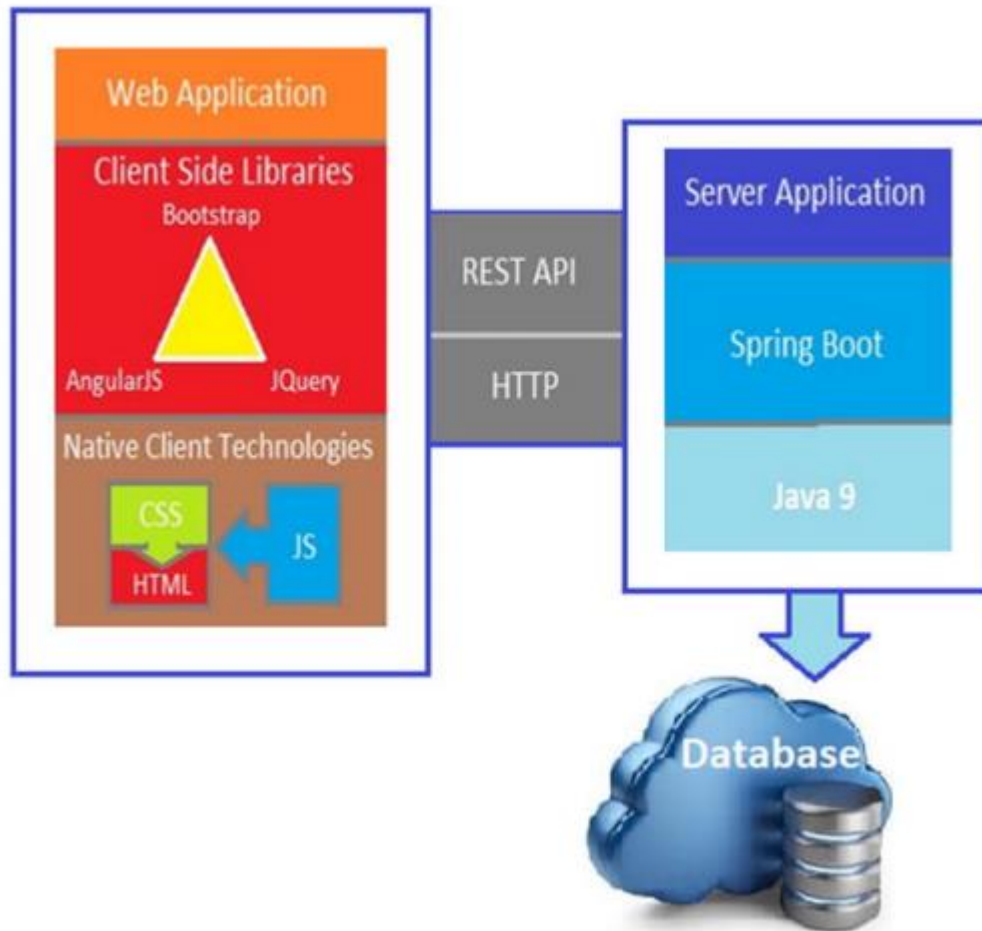


Figure2: Communication between the client and server

2.1.3 ARCHITECTURE OF MODERN WEB APPLICATIONS

In the beginning of the Web, web applications were non-existent. We had sites brimming with substance and still pictures. There were benefits and inconveniences to this.

Here are a couple of advantages:

1. There was too low a check header on the worker.
2. The substance itself was intensely packed as the information was stale and unaltered for most ISPs and worker suppliers.

Here are a couple of issues:

1. It was hard to refresh the substance.
2. There was nothing redone, and everybody had a similar encounter.
3. The UI used to be poor contrasted with the present use.

CGI has presented change by empowering dynamic substance utilizing composing dialects. On the drawback, it was on top of a very good quality PC, and it expected you to be a specialist in HTML and compose dialects on the worker side.

At that point came JavaScript, which makes pages more unique interestingly. It is principally utilized for fundamental composition to check structures or to show noticeable promotions. Improved convenience and decreased traffic stream to the worker with fundamental validation. Really awful you ought to have utilized the business thought twice, once on the customer side and once on the worker side. Subsequently, we have thought of the craft of present day web applications.

In this segment, I will examine the advancement of present day applications to conquer the fantasy that improving work area applications and applications is a decent approach. Current web applications center around dispersed foundation framework for content and previous customers on the Web utilizing RESTful APIs, as demonstrated in the fundamental design of the web application.

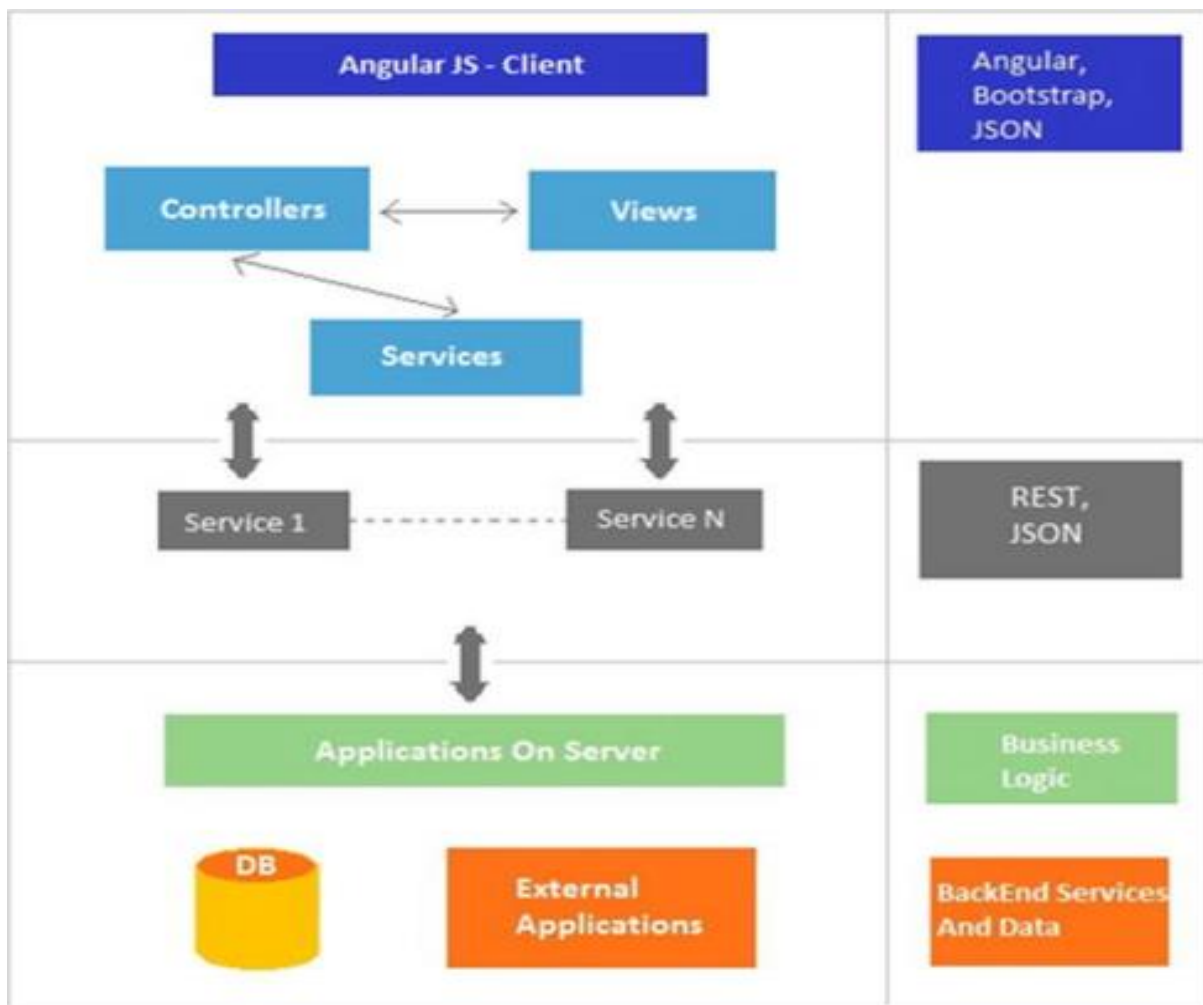


Figure 3: Layered Architecture of Modern Web Application

At the point when you need to collaborate with different articles on a page and those areas have a great deal of moderate things, at that point it is hard for all substance to utilize separate offers. Contrasted with customary next to each other engineering, the principle thought here is to construct a worker as a bunch of unused REST administrations. From the Model-View-Control View (MVC) see, the regulator is eliminated from the back end and moved to the customer. In this manner, the customer is presently ready to utilize the MVC design. The customer has a different view layer that contains the whole idea of the show. It likewise has a different pre-administration layer and control layer.

Toward the beginning of the application, just JSON information is moved among customer and worker utilizing the RESTful API. Each business thought will be revealed as REST endpoints to be utilized by Angular administrations.

The benefit here is that in the event that you have a versatile application for your web application, you can just utilize RESTful administrations without composing any extra code on the worker side of your portable application.

REST represents Transfer of State of Representation, which is an essential design style that diverges from the substantial item. It is a Web plan as it works today. You will become familiar with the APPLICABLE APIs exhaustively in Chapter 2 where you will make an invigorating layer of your application.

2.1.4 FRONTEND vs BACKEND

At the point when you make a website architecture, you partition the improvement cycle principally into two sections: the front and the backside. At the point when you talk about a web application or a portable application, you are fundamentally zeroing in on pre-extension improvement. At the point when you utilize any web application or any program, you type in a location, and the site opens with numerous things like content, pictures, structures, and remarkable intelligent highlights.

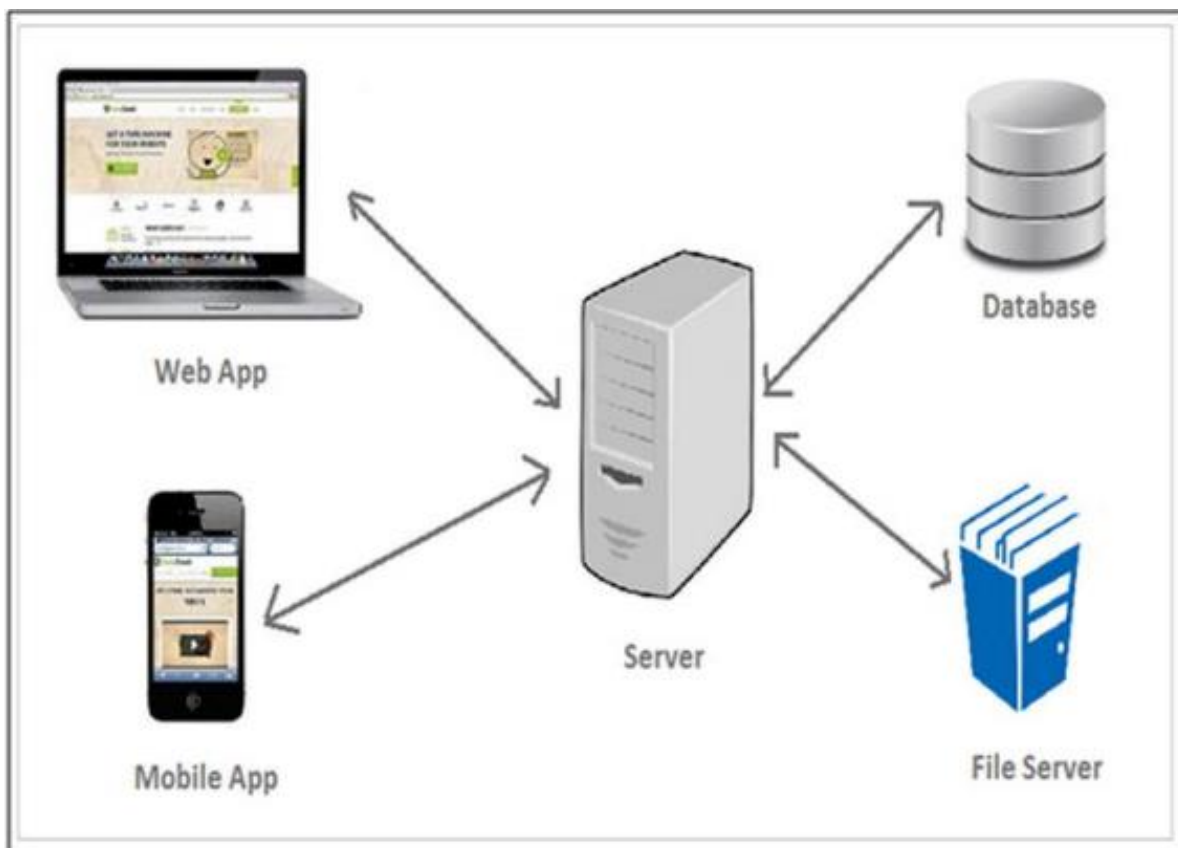


Figure4: Front and backend body view

On the off chance that you utilize a portable application, you will have an assortment of highlights, list sees, etc. These applications will show up on the customer side. Thus, regardless of whether you have a PC or a cell phone or are utilizing a program or versatile application, you will get the customer side of the total framework.

All client information is put away in a data set. All business thoughts and secure data are put away on a worker. The worker gets demands from the web application or portable application, measures these applications, recovers the necessary information from the data set or documents from the record worker, and sends them back to the past application. So essentially, the back stores the subtleties, and the front shows the subtleties.

Fashioners and engineers frequently utilize the system of prototyping sites and applications since building an application or site without any preparation takes a ton of time.

A product structure is characterized as a pre-characterized programming system for reusable modules that can be carried out to achieve a more effective and quicker errand. Thusly, the product system consistently contains libraries, compilers, and different APIs for fostering a particular marketable strategy. A decent full stack designer should know which structure is most appropriate for use in their arrangement of applications.

Here are a couple of famous programming parts accessible:

1. AngularJS (UI / Advanced)
2. Ionic (for hybrid mobile prototyping and development)
3. Pure MVC framework (application development framework)
4. Selenium (web exploration framework)

2.2 FRONTEND FRAMEWORK

The front end incorporates everything the client sees, including designs and different dialects like HTML and CSS. The primary reason for the front code is to speak with the client, just as to introduce the information in a distinct style.

There are many stunning JavaScript libraries that you can utilize when assembling a past application. Some of the time it very well may be interesting

to pick one single undertaking. The front end is every one of the things that cause a web framework look and to feel alluring. The past content is HTML, CSS and JavaScript. You will utilize AngularJS as a structure for additional improvement in this manual. I will cover AngularJS in more detail later.

2.2.1 HTML

There is no web without HTML. HTML is a tag language that describes the structure of web pages and determines how content is displayed online. HTML5 is the largest current version of HTML used. W3C (World Wide Web Consortium) is responsible for compiling HTML information followed by all browsers.

2.2.2 CSS

Cascading Style Sheets (CSS) is a style sheet language used to describe the presentation of a document written in markup language as HTML. CSS is designed to enable presentation separation and content, including textures, colors, and fonts. This segmentation can improve content accessibility, enable more flexibility and control in the presentation of presentation features, enable more web pages to share formatting by specifying the appropriate CSS in a separate .css file, and reducing the complexity and duplication of structural content.

2.2.3 BOOTSTRAP

Bootstrap is a front casing intended to make responsive plan a lot simpler. The Bootstrap CSS system can be utilized to style your site content. You can tweak

your CSS style to make your site look staggering, however Bootstrap offers an incredible arrangement of CSS styles that will permit you to make the most delightful substance plan. It isn't actually important to utilize Bootstrap while working with AngularJS, and there is no inner connection among AngularJS and Bootstrap as both are independent bundles. To utilize CSS from the Bootstrap CSS structure, you can characterize dependence on pom.xml while making the Spring Boot program for programmed download in the library envelope.

2.2.3 TYPESCRIPT

TypeScript is another language code that can be transferred to JavaScript. This modification is necessary because the browser cannot understand the generic code. TypeScript is a large typical JavaScript typed set that extends to JavaScript.

Transpiler converts the source code of one programming language to the source code of another programming language. Typescript brings JavaScript development closer to the traditional custom experience. TypeScript is based on ECMAScript 5 recommendations. Any JavaScript approved by TypeScript.

The features of TypeScript are:

1. Vertical Typing: Adds consistent typing to JavaScript, because the readability of the code improves and helps identify errors during compilation rather than detecting them during operation.
2. Module Support: TypeScript gives us the option to create modules, so that we can model the code for easy storage. The modules also help as they simplify the system.
3. Object Oriented Programming: TypeScript supports object-oriented programming elements such as category, folding, interface, asset etc. which helps to create more structured and reusable code.

4. Open Source: TypeScript is an open source. The Typescript source code can be downloaded from github.
5. Cross Platform: Works on all platforms like Windows, MacOS, Linux etc.
6. Support Support: TypeScript works best with Sublime Text, Eclipse and almost all major IDEs compared to JavaScript.

2.2.4 ANGULAR AS FRONTEND FRAMEWORK

AngularJS is a JavaScript-empowered library for web application advancement. Addresses the difficulties of a one-page framework (SPA). The AngularJS web application follows the MVC configuration design that prompts the formation of expandable, secure, tried, and suspended web applications. AngularJS information restricting and information make it an ideal accomplice for any worker innovation since it eliminates the vast majority of the code you ought to have composed, and everything occurs inside the program.

2.2.5 ANGULAR VERSION

AngularJS is all the more generally called Angular.js or now and then as AngularJS 1.x. AngularJS is a JavaScript-put together system based with respect to open source web application advancement and is profoundly kept up by Google.

Presently, Angular 4.0.0 is accessible and in reverse viable with 2.x.x for most applications. This new form contains some new highlights, for example, a back-up liveliness bundle that went into its bundle, an improved if support style if/else, and then some.

2.2.6 ANGULAR ARCHITECTURE CONCEPTS

You will currently take a gander at the ideas of AngularJS engineering. At the point when a HTML report is transferred to a program and tried by a program, the accompanying occurs:

1. The AngularJS JavaScript document is transferred, and an Angular worldwide item is made. A JavaScript record that registers manager capacities is made.
2. AngularJS filters HTML to see applications and perspectives on AngularJS and identifies task-related control capacities.
3. AngularJS utilizes control capacities and updates sees with information from the full control model.
4. AngularJS tunes in to program occasions, for example, button clicks, mouse developments, input field changes, etc. In case of any of these occasions, AngularJS will refresh the view in like manner.

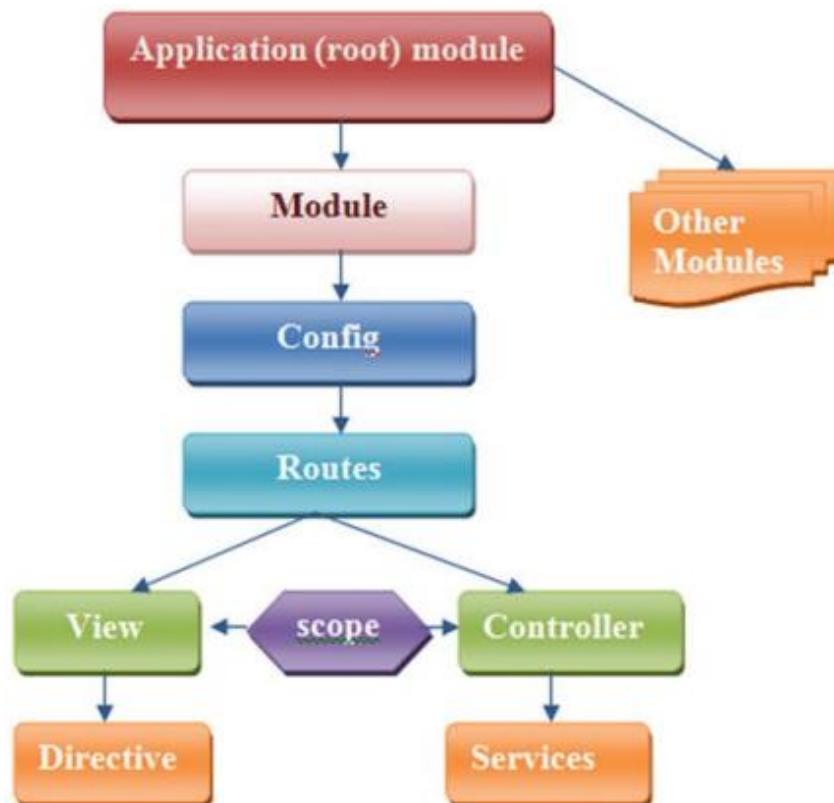


Figure 5 : Workflow Diagram Of Angular

AngularJS contains modules that fill in as holders for different sorts of uses like perspectives, managers, pointers, administrations, etc. The module determines how the application can be taken out from the boot. After that you have part of the config.

Courses are utilized to interface URLs with controls and perspectives. The idea is utilized to deal with a mind boggling occasion. Utilizations seeing pointers ng. The regulator controls the information for the AngularJS application, which contains standard JavaScript objects. AngularJS characterizes the ng-regulator bearing that makes new control components through the control work.

AngularJS accompanies a few underlying administrations like \$ http, \$ course, \$ window, \$ area, and the sky is the limit from there. Extension contains model-explicit components. They assume a significant part in getting the control together with a view. We will speak more about this later.

AngularJS utilizes MVC design to make web applications.

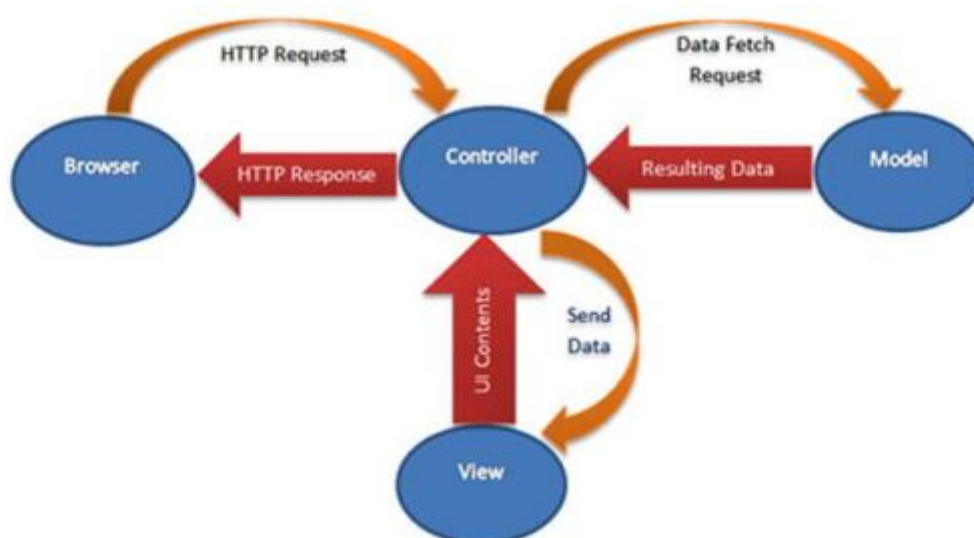


Figure6: MVC Architecture in Angular

At the point when a client sends a HTTP demand by means of a program, the solicitation is acknowledged by the regulator. The regulator measures that demand and sends the solicitation to the model to give the important subtleties. The reaction model gives the information game plan that shows up in the regulator also. The regulator measures the information again in the necessary configuration and sends it to see. Perspectives address information about UI content and send it to the regulator. At last, the regulator sends the HTTP reaction to the program.

1. The AngularJS see is utilized to create a portrayal of the yield of data, like a graph or chart, to a client in an internet browser. AngularJS makes sees in the report object (DOM) model by delaying every one of the layouts characterized by the application. Thus, the work of the engineer here is to simply make a layout utilizing more HTML and CSS.
2. The AngularJS model contains a \$ scope object used to store the application model, so there is no compelling reason to make a JavaScript model segment like other JavaScript customer side parts. The extension is connected to the DOM, assisting with making the JavaScript issue a lot simpler.

An AngularJS regulator is where you characterize all particular business ideas in a particular view. The regulator holds the model and looks together.

2.2.7 COMPONENTS

We know that angular improves single page applications. It doesn't mean we have to write all the app code in one place as it's not a good thing. That's why parts are used. They help us split the whole code into smaller pieces, keep all the pages and their business idea separately and upload it at the request of the user. Every Angular Application has one main feature and under this key element there are a few other sub-elements.

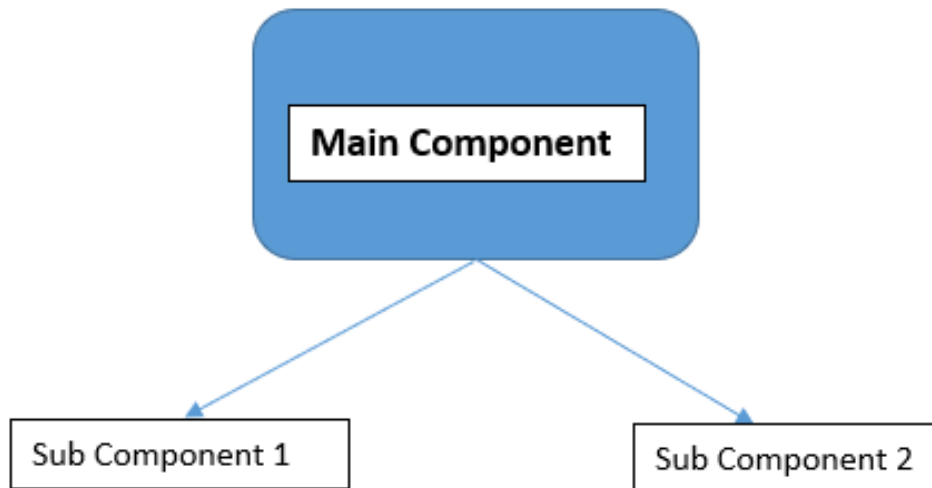


Figure7: Parts in Angular

We define the business concept of a component in an .ts file. This file will interact with the view using properties and paths, so we can say that the component controls the view. Angular creates and dissolves these ideas as a user continuity with the program. An item that is identified when a category has a @Component decor.

2.2.8 TEMPLATES

The templates represent an angular view of his role to show data whenever an event occurs. The HTML file associated with the section is known as the template. We can divide the templates into pairs:

1. Inline Template
2. External template

Inline Templates are defined within @Component decorator using template template. One should consider an inline template where the code is only two to three lines.

A file with the .html extension and linked in the corresponding section is called an external template. One has to use an external template where the code to be written is large and complex.

When we talk about templates we have to talk about their stuff. Template provides various features like translation, html, template statements etc.

2.2.9 BINDING DATA

In simple terms we can understand binding data as communication. The connection that takes place between TypeScript and viewing / html client sees on screen.

Angular offers a variety of communication methods between TypeScript and views such as one-way binding (view - part), one-way binding (view element), two-way binding.

Types of binding:

1. **Binding Properties** - Consider the example of a button, we want to disable this button, but how?
By disabling the button we will use the angular property defaults, here is how it binds the property.
Syntax: `<button property> click </button>`
2. **Responsible Responsibility** - By using this we can set the values of the viewing objects directly.
Syntax: `<h2 [attr.attribute] = "exp"> </h2>`
3. **Class Binding** - By using this we can add any css class to an object depending on the situation.
Syntax: `<div [class.classname]> </div>`
4. **Style Binding** - This is used to add middle styles to elements.

Example: `<button style = "background-color: green"> click </button>`

5. **Event Binding** - This binding is used to respond to user events for various purposes such as asking the employee to perform specific calculations etc.

Example: `<button (click) = "calculate ()"> calc </button>`, This will call the calculation method () whenever a button is pressed.

6. **Two-Way Binding** - Basically it is a group of goods and a binding event i.e. whenever the buildings are updated the UI will also be updated and vice versa.

We use `[(ngModel)]` to use binding in two ways.

2.2.10 DIRECTIVES

When an engineer wants to make Html attributes behave in a certain way or in a certain way, when guidelines are used. Guides are divided into 3 types:

1. Object instructions
2. Building instructions
3. Provide instructions

We've already talked about the order of things before.

STRUCTURAL DIRECTIVES:

They are responsible for changing the DOM system. They work by adding or killing components from DOM, not in any way similar to Attribute Directives that simply change the appearance of the part and lead.

Without a doubt you can distinguish between the Structure and the Attribute Directive by looking at the syntax. The name Structural Directive faithfully begins with an asterisk (*) prefix, whereas the Attribute Directive has no beginning.

Popular design instructions are:

1. ngFor - Just like for loop in any other programming language, ngFor is used to iterate or display data using loops.
2. ngIf - Similar to the statement if in Java or c ++, if it is executed if the given condition is satisfactory. It is used to make or finish a piece of DOM tree depending on the specific situation.
3. ngSwitch - Displays one item at a time from various other items to the statement of change. The only selected item included in the DOM is Angular.

ATTRIBUTE DIRECTIVES:

Provisioning guidelines change the look / functionality of an item / item. The following are built-in qualifications:

1. **ngStyle:** This guide is used to change object style / style. If there are more than one css styles you can use, we can use the ngStyle attribute. Example: [ngStyle] = "expression"
2. **ngClass:** This allows us to dynamically set and convert CSS classes to an object provided by DOM. If we have more than one CSS category to use, then we will go to Class. Example: [ngClass] = "expression"

2.3 BACKEND FRAMEWORK

The foundation code of any application can be considered as the mind of that application. The reinforcement code is made utilizing an outsider worker language and information and has never been seen by a client. The front code speaks with the end client continuously, and anything showed on the site inside the program is because of an inquiry made on the worker, where the foundation code meets the worker to recover client information before the end.

The manufacturer makes the last code toward the rear of the application utilizing a different worker code like Java, which associates the data set to save or refresh the information and afterward returns it to the end client as the past code. This sort of back-end code development assists designers with creating web based shopping applications, draw in with people in general, look for ongoing information, and accomplish more in the present Internet world.

There are numerous systems and libraries accessible that can be utilized to make backlinks a lot simpler and quicker. The most well known one you will use all through this book is the Spring Framework. The Spring Framework has consistently been well known in building kickback highlights for business applications, and with Spring Boot, the existence of an architect has never been so natural.

2.3.1 Java

Java is an elite, study hall based language, intended to depend on as not many use as could really be expected. It is a programming language proposed to accomplish normal targets expected to permit framework engineers to compose once, run anyplace (WORA), which implies that coordinated Java code can be applied to all Java support stages without the requirement for counter. Java applications are typically incorporated with bytecode that can run on any Java (JVM) machine paying little heed to fundamental PC arrangement. Java language structure is like C and C ++, however has less standard assets than some other. Java working framework gives dynamic capacities, (for example, show and change of working time code) that are frequently not accessible in incorporated local dialects. Since 2019, Java has

been perhaps the most broadly utilized dialects as per GitHub, particularly for customer web applications, with 9 million revealed engineers.

2.3.2 SPRING BOOT AS BACKEND FRAMEWORK

Why a Spring Boot? There are numerous systems for creating web applications, and Spring Boot is one of them. On the off chance that you need to fabricate something quicker, Spring Boot can be your best option as a web application advancement system.

The Pivotal group has fostered another lightweight system called the Spring Boot to work with the commencement and improvement of free applications for Spring and Production with a little drag that can "just run."

2.3.3 SPRING FRAMEWORK PROBLEMS AND SPRING BOOT ADVANTAGES

How about we see a portion of the issues of Spring Framework. The spring system was about the reliance immunization in 2006. It later developed and turned into an application structure that permits you to assemble Java marketable strategies. It tends to numerous business thoughts, for example, overseeing exchanges by giving layouts to make a straightforward strategy.

The Spring Framework additionally has a design module wherein Switzerland handles numerous normal concerns, for example, overseeing HTTP demands, information association, and so forth, and permits the designer to zero in on business administrations. The designer conducts business benefits and clarifies the classes with explanations given by Spring, advising Spring regarding those administrations. Spring likewise has framework backing, for example, interfacing with the RDBMS data set or the MongoDB information base.

Be that as it may, while conveying these highlights, Spring additionally brings different issues like a huge casing with numerous arrangement and design steps. There are numerous means to assemble and move.

To tackle these issues, Spring Boot ventured into the image. Spring Boot joins these means and permits designers to re-center around the business idea. The principle motivation behind the Spring Boot is to address the troubles of suspension in the Spring Framework by rethinking a large portion of the work to plan applications intended for Spring; no requirement for XML or coding more often than not.

Another energizing component offered by the Spring Boot is "no WAR, just JAR." Therefore, you don't have to produce a WAR document and transfer it to a Tomcat worker; you can make your own custom web applications and use them as Java JAR programs, making posting a lot simpler and more direct.

Spring Boot has thoughts, which means Spring Boot has a few bugs. You can make an application rapidly utilizing these profoundly utilized qualities. Spring Boot consequently decides required classes as per the libraries in its classpath.

Assume your application needs to communicate with an information base. Assuming there are Spring Data libraries in the homeroom, Spring Data naturally sets up data set associations and information source classification. Since Tomcat is a famous web holder, of course the Spring Boot web application utilizes the installed Tomcat compartment.

With Spring Boot, you can reveal things, for example, REST benefits freely, precisely as proposed in the development of microservices, so when you save any parts, you can presently don't utilize the whole framework.

With Spring Boot, you can disclose items such as REST services independently, exactly as suggested in the construction of microservices, so that when you save any parts, you can no longer use the entire system.

2.3.4) PRIMARY GOALS OF SPRING BOOT

Coming up next are the primary targets of the Spring Boot:

1. Provide creation prepared administrations and administrations with insignificant disturbance anybody can "use."
2. Ideas, which means settling on explicit choices for designers and the supporting extent of non-useful highlights regular in business frameworks (inserted workers, security, wellbeing testing, measurements, and outer arrangement)
3. Support incorporation over suspension, keep away from XML suspension out and out, and ensure explanation suspension
4. Allowing designers to redo Spring Boot programs the manner in which they like.

2.3.5 ANNOTATIONS OF SPRING BOOT

The principal comment of the `HelloSpringBootApplication` class is `@SpringBootApplication`. A `@SpringBootApplication` explanation is a basic comment presented in Spring Boot 1.2.0 and adds the accompanying comments:

1. `@Configuration`: The `@Configuration` explained segment can be utilized with the Spring holder as a wellspring of bean definitions, not indicated in Spring Boot. This part may contain at least one bean ads in the structure determined in the `@Bean` comment.
2. `@EnableAutoConfiguration`: This explanation is important for a Spring Boot project that reveals to Spring Boot to begin adding beans utilizing classpath portrayals or settings. The programmed completing change predicts and readies the beans that you may use with the framework and along these lines improves on crafted by the engineer.
3. `@ComponentScan`: This explanation advises Switzerland to search for explicit bundles to filter explicit segments, arrangements, and administrations.
4. `@RestController` and `@RequestMapping` Annotations The primary explanation of the `HelloSpringBootApplication` class is `@RestController`, which is a comment of strange notion. The `@RequestMapping`

explanation gives "course" subtleties and educates Switzerland that any HTTP demand with a way/hi ought to be set apart in the hello way.

Explanations for `@RestController` and `@RequestMapping` are from Spring MVC (these definitions are not determined in Spring Boot).

2.3.6 CONTROLLERS IN SPRING

In order to use administrators in our business plans we need to add spring-webmvc dependencies to the pom.xml file. The control section helps manage the application submitted by the client. When a request is received the controller uses the appropriate business concept to process that request and redirect the client to the view to deliver results.

If the category has an `@Controller` annotation, this means that the next section is the control section.

`@ResponseBody` - states that all control systems have their own return values written directly in response, instead carried by model for delivery.

The combination of these two definitions is provided by `@RestController`.

The controller provides an overview of how to manage maps with specific HTTP requests. We will discuss the following definitions:

1. `@GetMapping` - This annotation is used to manage an HTTP GET application. In simple terms we can say that downloading data from the `@GetMapping` database is in use.
2. `@PostMapping` - This annotation handles the HTTP POST request. In simple terms we can say that entering data in the `@PostMapping` database is in use.
3. `@PutMapping` - This annotation handles the HTTP PUT request. Used to update the value in the database.
4. `@DeleteMapping` - This annotation handles the HTTP DELETE request.

2.3.7 SERVICE LAYER USING SPRING

In any business application service layer its heart. It contains all the business ideas needed for the smooth running of a program. This layer deals with the presentation layer and persistence layer of the application.

The business application layer layer is the layer where the business idea is inserted. It is the heart of business applications as it contains in-app business processes. It works with the presentation layer and the background layer. In this tutorial, you will learn how to improve the application layer of business applications using Spring Boot.

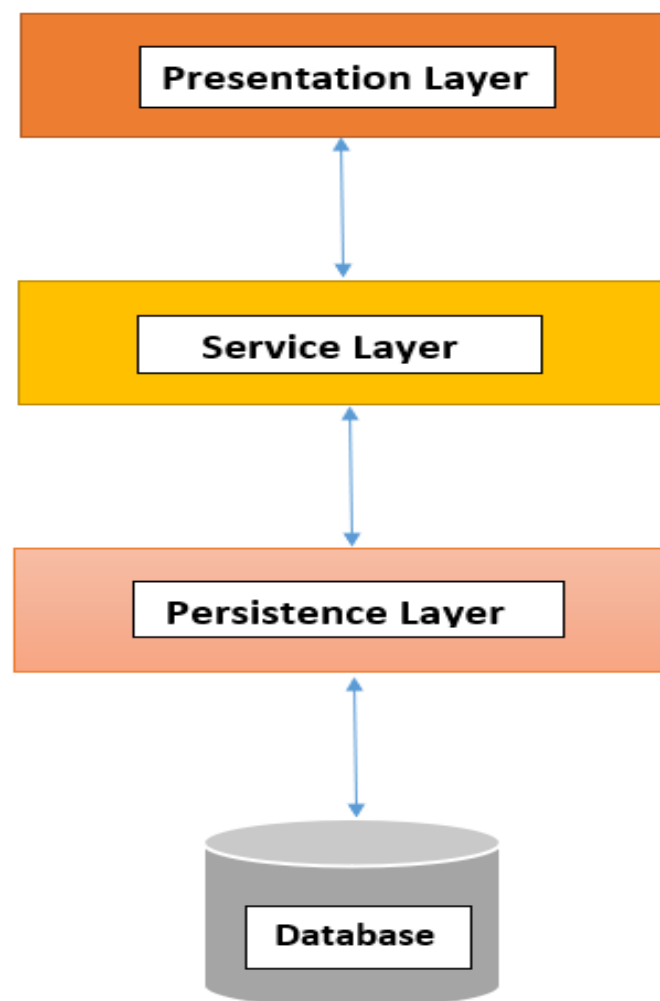


Figure8: Various Spring Layers

2.3.8 PERSISTENCE LAYER USING SPRING

The persistence layer integrates with the relationship database and the service layer. Receives data from a service layer, performs functions in a database and sends results back to the application layer. The database interaction code is applied to this layer. In this lesson, you will learn the development of a persistence layer using Spring ORM and Spring Data.

Spring is a popular framework for the development of business applications. By using a persistent layer, it offers many options. The developer can select appropriate Spring modules such as Spring JDBC or Spring ORM to integrate with JPA, Hibernate etc. in using the enterprise data access layer.

Spring continues to facilitate the development of a persistent crust using Spring Data, which is an umbrella project that supports relationship-related and unrelated information. Provides a continuous continuous layer code with minimal codes from the developer thus eliminating the great effort of the developer.

The persistence layer acts as a mediator between the database and the service layer. The service layer provides data in the persistence layer, the persistent layer uses that data to perform tasks in the database and returns the results to the service layer. (Persistence layer contains a database connection code).

We can improve this layer using Spring ORM.

2.3.9 SERVICES

Services help us build a renewable and sustainable code. Angular doesnot provides for any special decoration of the service category, we just need to

define the category and apply it where needed using a dependency injection. The service component is called {name}.service.ts.

While using an angular application we need to connect our application to go back and connect with it, this can be done using the HTTPClientModule. We must import it from @angular/common/http. The great advantage of using this is that when our app connects to the backend, and the application is still being processed our screen is still interactive which is not cold. We can use HTTP to execute POST, PUT, GET and DELETE commands.

When we apply using HTTP, the application goes backwards and then returns the results back as a cluster. This acquisition of a collection at some point is known as visual. Visually produces data that a user can register to use. How do we register?

It is much easier to use the subscribe () function. Let's understand by using an example considering that there is a dairy company that delivers milk to each family you subscribe to. Only customers registered with this company will receive their product. It is now up to the dairy owner to decide when the customers will receive the next bottle of milk. It is not necessary for a whole bottle of milk to be delivered at the same time. Customers may need to wait for the next bottle to arrive.

Visual enhances the performance of the angular system. As they facilitate differential communication and when requests and responses are sent concurrently the performance increases.

2.3.10 CROSS ORIGIN RESOURCE SHARING

If we try to connect our backend with the frontend and connect it, it will work in a systematic way. There will be an error in the console for "request blocked due to CORES". Without using the CORES server it will not send any feedback back before.

The `crossorigin` attribute sets the application mode to the HTTP CORS application. Web pages often make requests to upload resources to other servers. This is where CORS comes in.

Cross-Origin application is a resource application (e.g. style sheets, iframes, images, fonts, or text) from another domain.

CORS are used to manage cross-cutting applications. CORS stands for Original Resource Sharing, and is a method that allows resources on a web page to be requested from a domain other than t

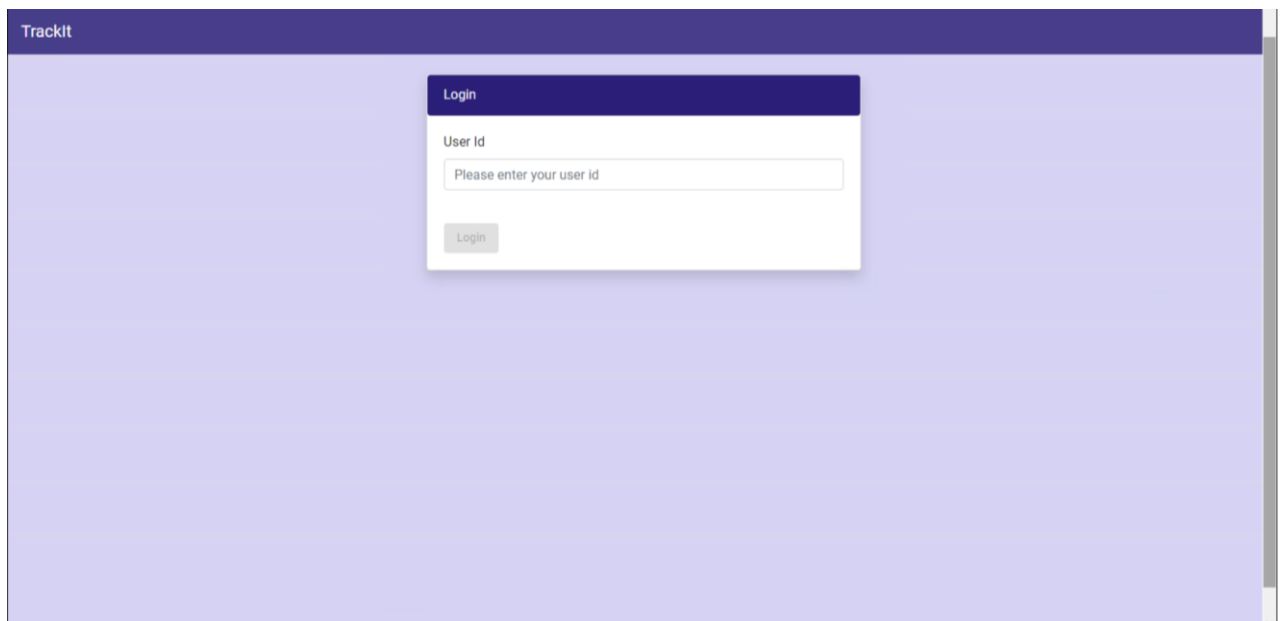
their background. Describes how the browser and server can work together to determine whether it is safe to allow a deviant request. CORS allows servers to specify who can access assets on a server, among many other things.

CHAPTER 3

IMPLEMENTATION WORK

User Story 1 : As a trainee/batchowner one should be able to login to our respective pages automatically just by entering the user id.

Login Page:



TrackIt

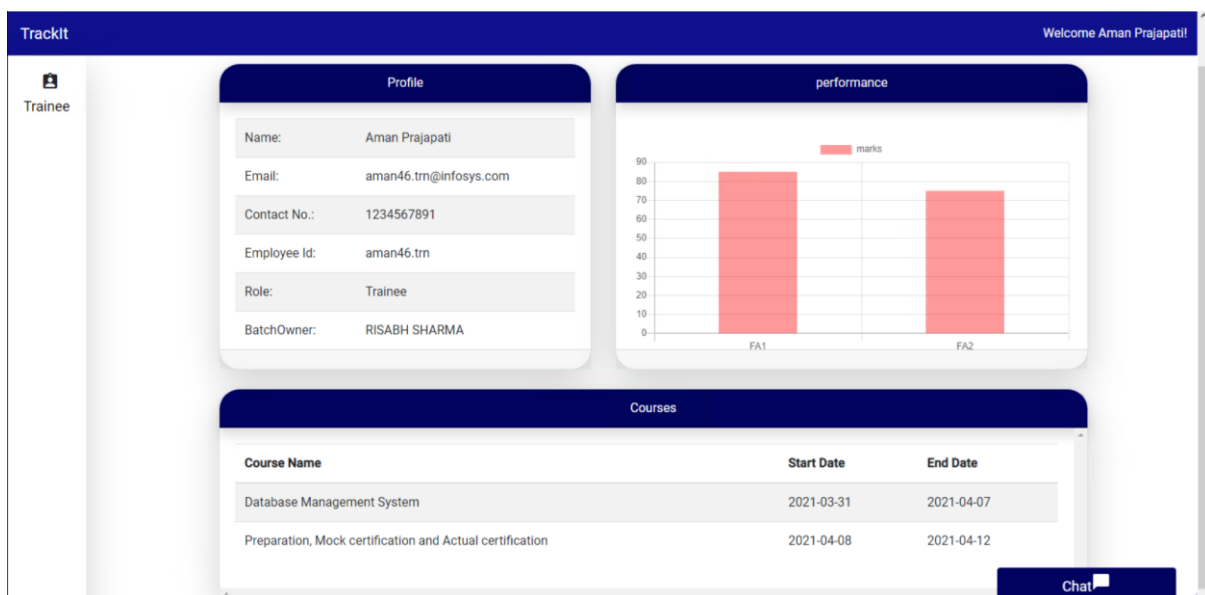
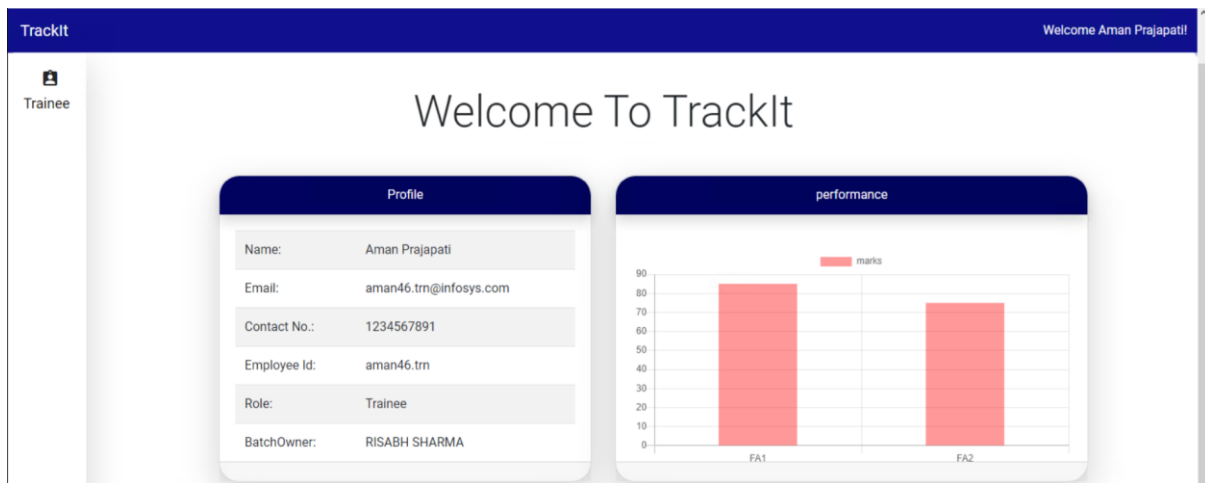
Login

User Id

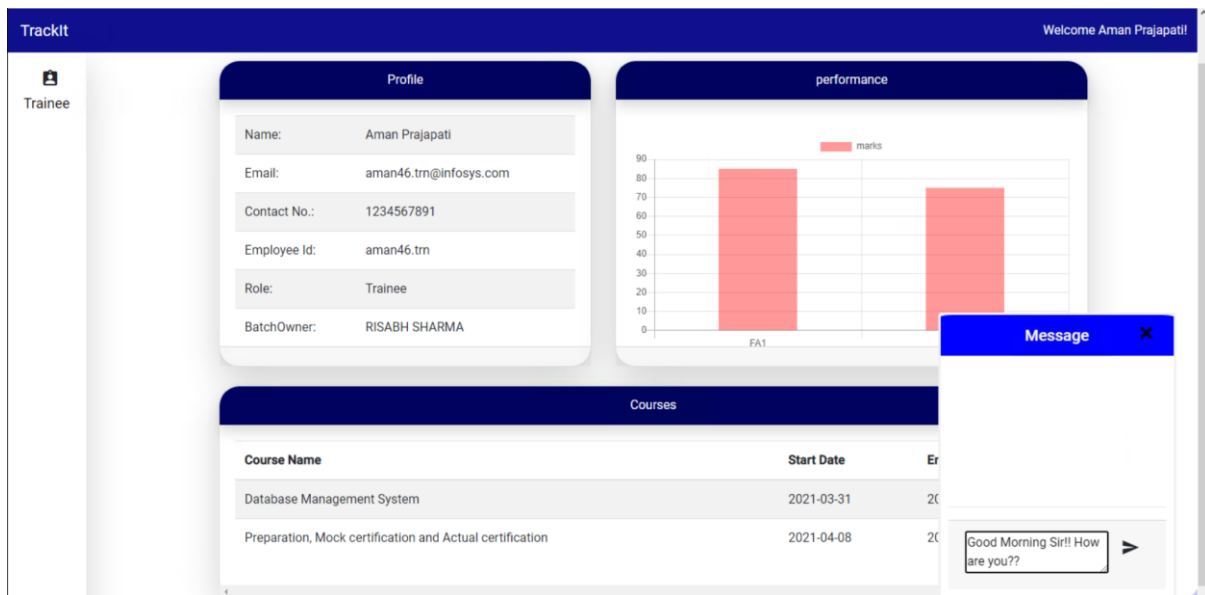
Please enter your user id

Login

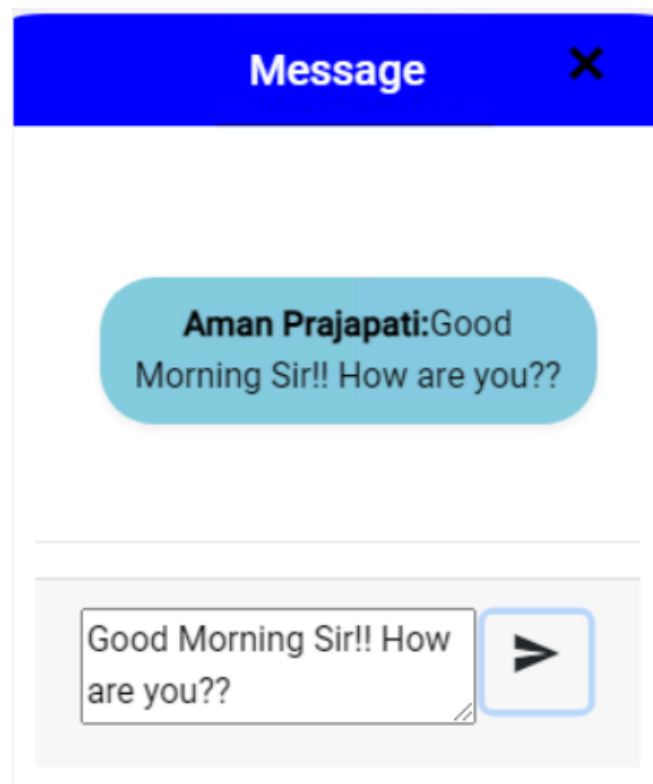
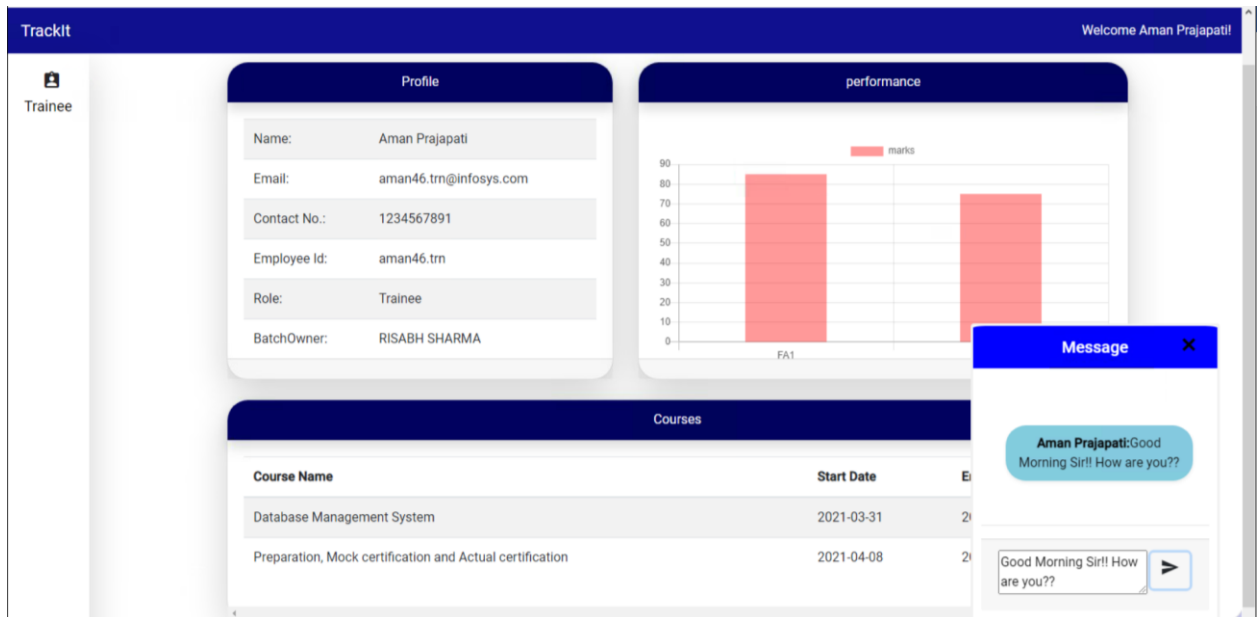
User Story 2: As a trainee one should be able to view our personal details along with our academic performance.



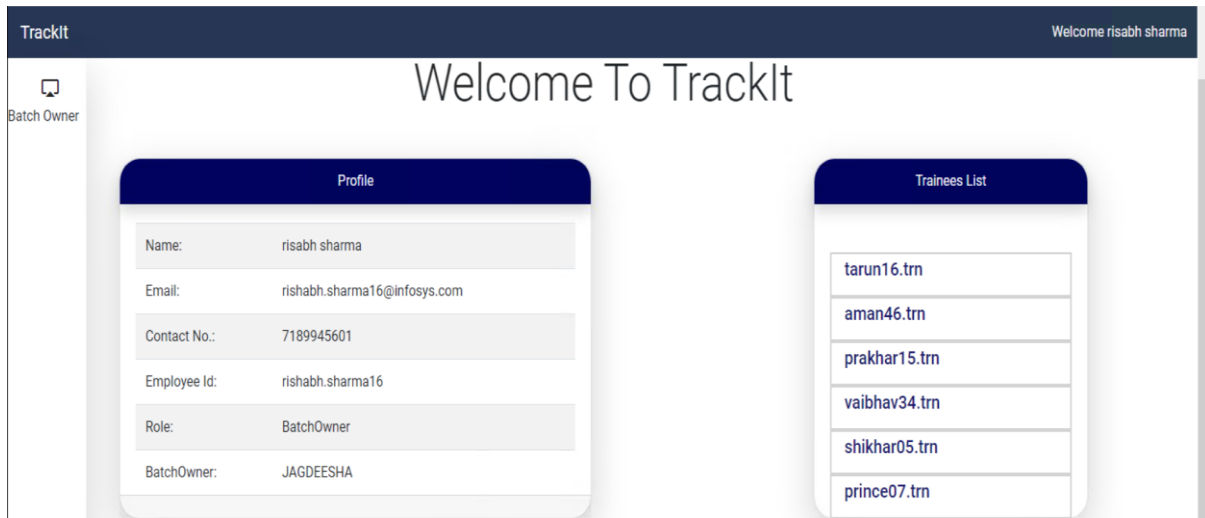
User Story 3 : As a trainee one should be always connected to our batchowner directly.



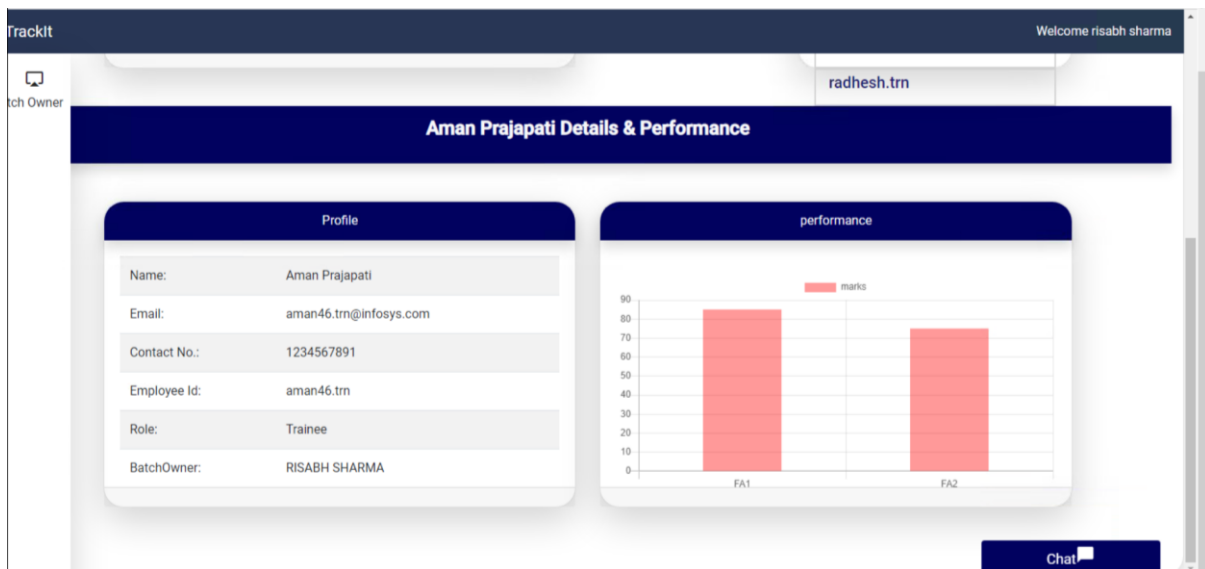
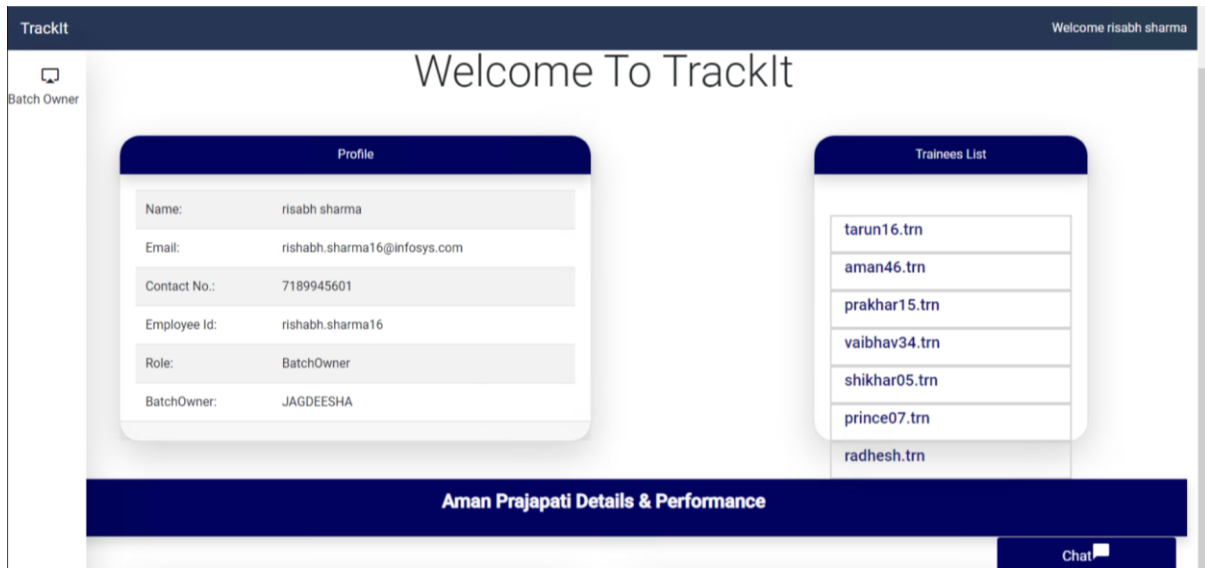
User Story 4: As a trainee one should be able to put queries to the batchowner and also view the past queries put by me.



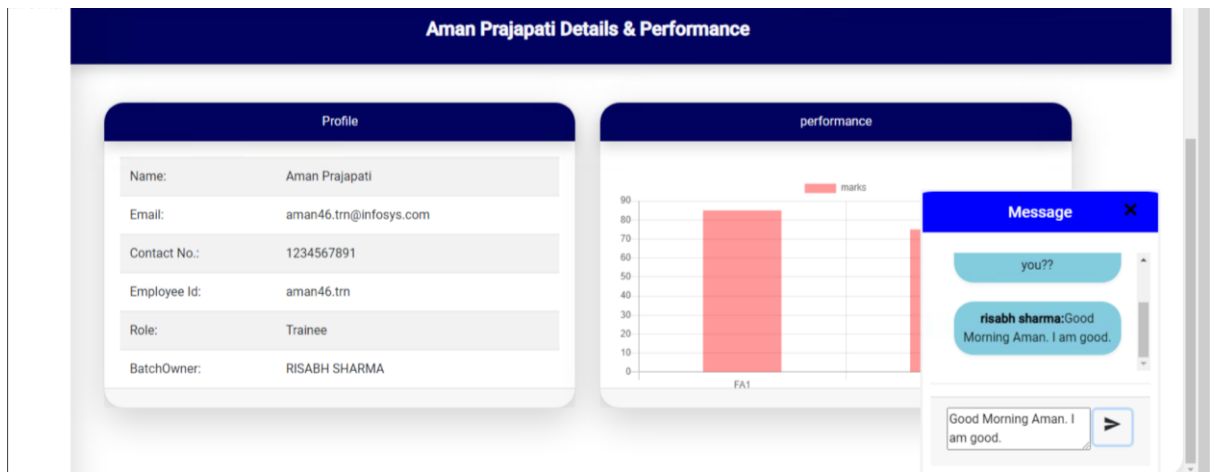
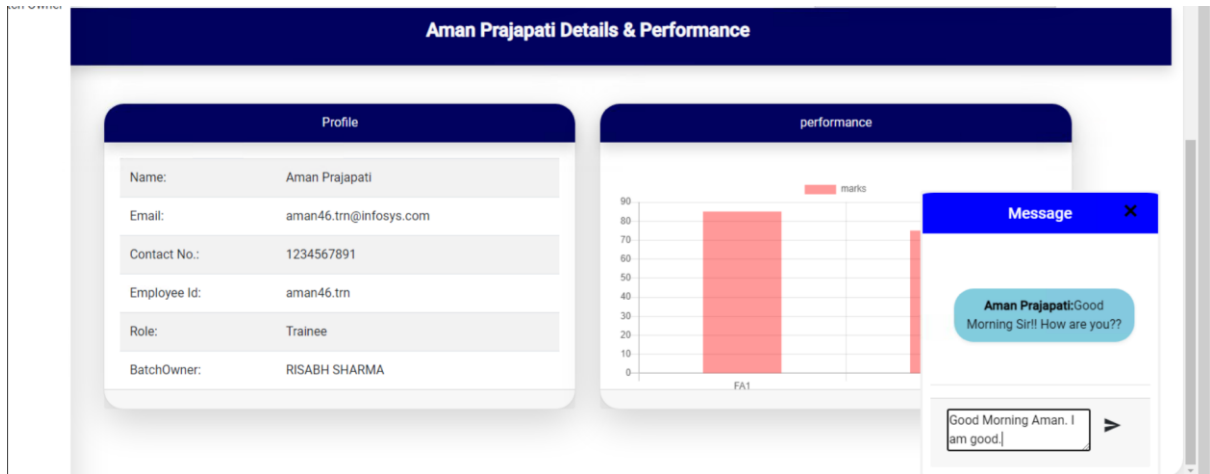
User Story 5: As a Batchowner one should be able to view my profile.



User Story 6: As a Batchowner one should be able to view the details of trainees under me.



User Story 7: As a Batchowner I should be able to see the queries put to me by my trainees and reply to them.



CHAPTER 5

RESULT

We have successfully implemented the three problem statements mentioned below :-

1. As a trainee/batchowner one should be able to login to our respective pages automatically just by entering the user id.
2. As a trainee one should be able to view our personal details along with our academic performance.
3. As a trainee one should be always connected to our batchowner directly.
4. As a trainee one should be able to put queries to the batchowner and also view the past queries put by me.
5. As a Batchowner one should be able to view my profile.
6. As a Batchowner one should be able to view the details of trainees under me.
7. As a Batchowner I should be able to see the queries put to me by my trainees and reply to them.

References

- [1] "Spring Boot Tutorials | Java Development Journal", *Java Development Journal*, 2021. [Online]. Available: <https://www.javadevjournal.com/spring-boot/>. [Accessed: 18- May- 2021].
- [2] V. Balasubramaniam, "Defining JPA Entities | Baeldung", *Baeldung*, 2021. [Online]. Available: <https://www.baeldung.com/jpa-entities>. [Accessed: 18- May- 2021].
- [3] J. Journal, "Spring Boot Annotations", *Java Development Journal*, 2021. [Online]. Available: <https://www.javadevjournal.com/spring-boot/spring-boot-annotations/>. [Accessed: 18- May- 2021].
- [4] "The Spring @Controller and @RestController Annotations | Baeldung", *Baeldung*, 2021. [Online]. Available: <https://www.baeldung.com/spring-controller-vs-restcontroller>. [Accessed: 18- May- 2021].
- [5] "Angular", *Angular.io*, 2021. [Online]. Available: <https://angular.io/guide/architecture-components>. [Accessed: 18- May- 2021].
- [6] "Angular", *Angular.io*, 2021. [Online]. Available: <https://angular.io/api/common/NgIf>. [Accessed: 18- May- 2021].
- [7] "*ngFor Directive in Angular | DigitalOcean", *DigitalOcean*, 2021. [Online]. Available: <https://www.digitalocean.com/community/tutorials/angular-ngfor-directive>. [Accessed: 18- May- 2021].
- [8] "Angular", *Angular.io*, 2021. [Online]. Available: <https://angular.io/api/common/NgSwitch>. [Accessed: 18- May- 2021].